OIPE JC95
SEP 11 2003
PATENT & TRADEMARK OFFICE

RECEIVED
SEP 15 2003
Technology Center 2100

REMARKS

Claims 1-10 are pending in the current application. In the Office Action dated May 5, 2003, the Examiner rejected claims 1, 2, 4-6, and 8-10 under 35 USC §103(a) as being unpatentable over Wirthlin et al., "The Nano Processor: a Low Resource Reconfigurable Processor" ("Wirthlin") in view of Que's Computer Users Dictionary, Fifth Edition., Pfaffenberger, Brian, 1994 ("Que"), and Page, "Reconfigurable Processors" ("Page"), rejected claim 3 under 35 USC §103(a) as being unpatentable over Sudo, U.S. Patent No. 6,047,198 ("Sudo") in view of Wirthlin, Que, and Page, rejected claim 7 under 35 USC §103(a) as being unpatentable over Sudo in view of Wirthlin and Page, and rejected claims 1-10 under 35 USC §103(a) as being unpatentable over Huffener, U.S. patent No. 5,382,891 in view of Wirthlin. Applicants respectfully traverse all of the above-mentioned 35 USC §103(a) rejections.

Applicants' representative wishes to thank the Examiner for removing an earlier final rejection, and again considering the current application. While Applicants' representative values a thorough examination, Applicants' representative believes the current rejections are somewhat overly complex and cover relatively little new ground, although Wirthlin and Page indeed provide additional insight as to the state-of-the-art background existing at the time the current application was filed. The rejections are overly complex, because certain of the cited references add nothing beyond the Background of the Invention section included in the originally filed application, as is discussed below. The cited references either were previously cited, are related both in subject matter and authorship to previously cited references, or, in the case of Huffener, completely irrelevant, providing no additional support for rejections of the claimed invention.

The Office Action contains many seemingly incorrect statements and inferences, not all of which are addressed in this response. For example, the statement "Using the complex programmable logic device/micro-controller IC system would have allowed for less glue logic such that LCD control would have been part of the overall system control, or CPU, rather than extra, external circuitry" is self-contradictory, and doesn't make much sense to Applicants' representative. The claimed invention *is* a

subsystem controller, not a CPU. Similarly, the statement "The bus interface as taught by Huffener must be by way of I^2C, or inter-integrated circuit buses such that any bus connecting 2 integrated circuits is an I^2C bus" is incorrect. The I^2C bus is a specific, industry-standard communications technology that has been used for over 20 years, and is extremely well-known to electronics and systems engineers. However, in the interest of brevity, Applicants focus, below, on a number of major points relevant to the rejections.

Wirthlin

The Examiner states in the Office Action on page 3, in section 6, "Wirthlin et al. have also not taught that the subsystem controller is implemented as a single integrated circuit." As discussed in previous responses, Applicants are clearly and ambiguously claiming a "subsystem controller implemented as a single integrated circuit for control of a device or subsystem within an electronic system having system processing components." Wirthlin does disclose employing a field programmable gate array, *rather than a standard microprocessor*, to implement a processor, used, in one example, within something similar to a subsystem controller. For example, in the first and second paragraphs of section 3, on page 2, Wirthlin discloses:

> The Nano Processor (nP) is a stored-program processor that achieves application-specific performance with general purpose programmable control. The nP implements application-specific functionality through the development of custom instructions. An integrated assembler generates the program data necessary to convert custom assembly instructions into executable code.
>
> Similar to the Reconfigurable Microprocessor[6], the nP implements the processor control within a FPGA instead of using a standard microprocessor. (emphasis added)

In the Abstract, Wirthlin describes:

> One way to introduce a more flexible development approach is to implement a customizable stored-program processor. For a given application, the designer can develop customized hardware to increase performance and then control the sequencing and operation of this hardware with software.

However, Applicants have already discussed a range of subsystem controller implementations in the current application. For example, beginning on line 24 of page 1 of the Background of the Invention section of the current application, in

discussing subsystem controllers available at the time that the current application was filed, Applicants clearly stated that within existing subsystem controllers:

> control functionality is partitioned between logic circuits, including one or more programmable logic arrays ("PLAs") or programmable logic devices ("PLDs"), and software routines stored in an electronic memory and executed by a micro-controller. Figure 1 graphically illustrates a continuum of possible partitionings of functionality between hardware and software. In general, the control functionality of a subsystem controller can be entirely implemented in hardware, for example as an application specific integrated circuit ("ASIC"), as indicated in Figure 1 by the ASIC and logic circuitry 102 at the left end of a line 104 representing a continuum of possible hardware/software partitionings within a subsystem controller. On the other hand, the control functionality within a subsystem controller may also be entirely implemented as software routines running on a micro-controller, as indicated in Figure 1 by the micro-processor 106 at the right side of the continuum. Normally, control functionality is partitioned between hardware and software, as indicated by the subsystem controller 108 positioned towards the center of the continuum. For any specific subsystem controller, a decision is normally made as to the partitioning, and appropriate components are selected and together integrated on a printed circuit board ("PCB"). (emphasis added)

Wirthlin adds nothing new. Wirthlin describes an approach fully comprehended within the above description of existing subsystem controller implementations.

Sudo

As stated by the Examiner in the first sentence of page 5 of the Office Action, "Sudo has not taught the subsystem controller of claim 1." Applicants' representative discussed Sudo in a previous response:

> However, again, Sudo does not disclose a single-integrated-circuit implementation of a subsystem controller. Instead, in the figures referred to, and cited by, the Examiner, Sudo shows a standard collection of discrete components included within a computer system. As one example, the Examiner asserts that element 5A of the Figure 4 in Sudo teaches a controller programmed to display information. In fact, as noted in Sudo on lines 13-15, the box labeled 5A in Figure 4 is a liquid crystal display driver, or, in other words, a set of software routines that are run on the CPU to control the LCD 5. Element 5A of Figure 4 is not a controller programmed to display anything. Instead, it is a program. Sudo does not disclose a subsystem controller, because, as pointed out in the Background of the Invention section of the current application, a subsystem controller is a processing component independent of the CPU of the system, that, in part, functions to offload computing tasks from the central processing unit of

the system (current application, page 1, lines 19-23). Sudo, by contrast, discloses a software control program that runs on the central processing unit of the system.

The Examiner further states:

> An artisan would have been motivated to employ the system of Wirthlin et al., Que, and Page, as the CPU of Sudo, where the complex programmable logic device is used as an input/output controller, in this case the LCD controller.

Were an artisan to do this, the artisan would have replaced Sudo's original CPU with some kind of hybrid processor, but that hybrid processor would still not constitute a subsystem controller, as clearly defined in the current application and in claim 1:

> A *subsystem controller* implemented as a single integrated circuit for control of a device or subsystem *within an electronic system having system processing components*, the subsystem controller comprising:
>
> a complex programmable logic device that can be programmed to provide logic circuits that implement control functionality;
>
> a micro-controller that can execute software routines that implement control functionality;
>
> read-only memory that stores executable code for execution by the micro-controller;
>
> random-access memory that can store data and executable code for execution by the micro-controller;
>
> a bus interface for exchanging data and control signals between the subsystem controller and system processing components; and
>
> an additional electronic interface to a device or subsystem controlled by the subsystem controller. (emphasis added)

Please note that claim 1 clear claims a subsystem controller that is a separate and distinct entity from a system processor, and please note further that claim 1 is consonant with the definition of a subsystem controller provided in the current application, beginning on line 11 of page 1:

> Subsystem controllers are ubiquitous components of modern computer systems, peripheral devices within computer systems, and other electronic devices. The term "subsystem controller" generally refers to a subcomponent of a more complex electronic system, such as a computer, that comprises logic circuits, a programmable logic device, and a general-purpose micro-controller that executes a number of software routines. A subsystem controller is generally dedicated to one or a small number of specific control tasks. For example, the control of LED and LCD display devices incorporated in a front panel display of a computer system is generally carried out by one or more subsystem controllers. Use of subsystem controllers may offload computationally

> intensive and time-intensive tasks from the main processor or processors of computer systems, and may significantly decrease data traffic on critical busses of the computer system that are bottlenecks for data movement within the computer system. (emphasis added)

Sudo, like Wirthlin, adds nothing to the discussion that was not already either explicitly stated in the Background of the Invention section of the current application, or understood by those familiar with computer systems and computer system architecture.

Que

The Examiner, in section 5 of the Office Action on page 3, states that:

> Que has taught utilizing ROM in place of random-access memory (RAM) to store executable code to be run on the micro-controller would have allowed for the system to retain such code in the system power-off state and load the code immediately upon a change to power-up for execution by the micro-controller (Que page 146). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to utilize read-only memory, or ROM, to store executable code for execution by the micro-controller as taught by Que instead of random-access memory, or RAM, of Page, in order to ensure immediate loading of the code upon system power up.

Que teaches nothing of the sort. Que describes a trend of storing portions of an operating system on ROM chips instead of on disk. Moreover, Que makes it clear that these are startup instructions and functions, needed to bootstrap a computer system:

> Because the computer's random-access memory (RAM) is volatile (loses information when you switch off the power), the computer's internal memory is blank at power up, and the computer can perform no functions unless given startup instructions. These instructions are stored in ROM. A growing trend is toward including substantial portions of the operating system on ROM chips instead of on disk.

The Examiner's statement mischaracterizes Que. Moreover, Que is discussing computer systems, and discusses discrete RAM and ROM chips within a computer system. Page discusses a processor. Que does not suggest including either RAM or ROM within Page's processor, or within any kind of integrated circuit that includes functionality other than memory.

Huffener

Huffener is a completely irrelevant reference. The Examiner states, in section 22 of the Office Action, that "Huffener has taught a subsystem controller implemented as a single integrated circuit for control of a device or subsystem within an electronic system having system processing components (Figure 12)." A cursory review of Figure 12 reviews a quite contrary implementation. As discussed beginning on line 9 of column 10 of Huffener, the logic circuit shown in Figure 12 includes discrete encoders 94 and 95 interfaced with a discrete "8-bit single chip microcontroller" 85, and numerous other components. As discussed by Applicants' representative in a previous response:

> In The Electrical Engineering Handbook, Dorf, Richard, CRC Press, 1993, page 614, an integrated circuit or IC is defined as: "an assembly of miniature electronic components simultaneously produced in batch processing, on or within a single substrate, which performs an electronic circuit function." In other words, an integrated circuit is what is commonly referred to a chip. A collection of discrete components soldered to, and electronically interconnected by, signal lines on a printed circuit board is not an integrated circuit.

A logic circuit that includes a discrete "8-bit single chip microcontroller" to which other components interface is – by the above definition, and by the definition of anyone remotely familiar with computer system and electronic – not a single integrate circuit. Like Wirthlin and Sudo, Huffener is irrelevant to the claimed invention. Huffener's device is not a subsystem controller, as clearly defined in the current application, and as clearly claimed, as discussed above with respect to Sudo.

Page

Applicants' representative acknowledges that Page is useful background information with respect to the current application – the only relevant reference of the five cited references. Page discloses combining a processor and a dynamically programmable gate array within a single integrated circuit. Page is, however, clearly concerned with implementing CPUs, not subsystem controllers. For example, Page argues, in the second paragraph on page 1, that:

> conventional microprocessors are a poor match for any *particular* application, though they support a wide range of applications well. We suggest that the solution is to use additional silicon to provide hardware which can be configured

> to support any application. By combining a conventional processor with a DPGA on a single chip, commodity pricing is maintained and yet the same part can be targeted effectively across a wide range of applications.

Subsystem controllers do not run applications, and certainly do not run *any* applications. Page is clearly proposing a type of general purpose CPU – not a subsystem controller. Later, Page states:

> However, there is scope for real technical innovation since there is no longer the same necessity to regard the processor core as sacrosanct and we can consider making changes to it as well as to the DPGA so that the combination becomes even better at supporting user applications. (emphasis added)

Page describes a first, non-IC implementation of his processor, in the last full paragraph of page 3:

> The HARP system goes a little beyond the simple connection of an DPGA to a microprocessor bus in that it provides a two banks of SRAM connected to the DPGA. This memory is there as a general-purpose high-speed memory for storage of data or temporary results associated with the co-processor computation. By altering the DPGA configuration, this memory can be used privately by the DPGA algorithm, or it can be mapped onto the microprocessor bus where both processors can access it.

Furthermore, in section 6.1, Page again indicates that the reconfigurable processor "contains any microprocessor with a relatively fixed architecture which uses DPGA technology within one or more of its major architectural modules," and lists other areas in which DPGA support might be included, including the ALU, register file, memory interface, DMA, device interfaces, instruction decode, floating-point processor and goes on to prose various projected uses and embodiments. Nowhere in Page is there a mention or suggestion of incorporating "read-only memory that stores executable code for execution by the micro-controller," "a bus interface for exchanging data and control signals between the subsystem controller and system processing components," and "an additional electronic interface to a device or subsystem controlled by the subsystem controller." A read-only memory is not contemplated by Page, because Page's disclosed processor is not a subsystem controller, "dedicated to one or a small number of specific control tasks," but is instead a general purpose CPU, intended to be included in a larger system with discrete components for storing bootstrap code, as described by Que. A subsystem controller, by contrast, uses a "read-only memory that stores executable code,"

both because a subsystem controller generally configures itself, rather than loading programs from disk into memory, and because, since it is highly specialized, the relatively few software routines needed for control of a subsystem controller can fit within a ROM. By contrast, the software applications and operating systems run by a general purpose CPU cannot possibly fit within ROM – at least – a commercially feasible ROM.

Summary

Consider claim 1:

> 1. A *(1)* subsystem controller *(2)* implemented as a single integrated circuit *(3)* for control of a device or subsystem *(4)* within an electronic system having system processing components, the subsystem controller comprising:
>
> *(5)* a complex programmable logic device that can be programmed to provide logic circuits that implement control functionality;
>
> *(6)* a micro-controller that can execute software routines that implement control functionality;
>
> *(7)* read-only memory that stores executable code for execution by the micro-controller;
>
> *(8)* random-access memory that can store data and executable code for execution by the micro-controller;
>
> *(9)* a bus interface for exchanging data and control signals between the subsystem controller and system processing components; and
>
> *(10)* an additional electronic interface to a device or subsystem controlled by the subsystem controller. (emphasis and parenthesized numerals added)

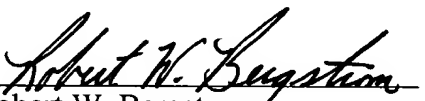
As discussed above, only a single cited reference, Wirthlin, seems to disclose (1)(3)(4)(9)(10) a subsystem controller, but fails to disclose or suggest (2) a single-IC implementation, (6) a micro-controller, and a (7) read-only memory. Page discloses a single-IC implementation, but of a general-purpose processor, rather than a subsystem controller, including (5) a complex programmable logic, (6) microcontroller, and (8) RAM. Que provides brief, general definitions of RAM and ROM, but not in the context of subsystem controllers. Neither Huffener nor Sudo relate to subsystem controllers, but mention (6) and (8) in different contexts. In Applicants opinion, there are simply too many claim elements that are not disclosed, mentioned, or suggested in the cited references. In particular, Applicants are claiming a subsystem controller implemented as a single integrated circuit, and there is simply no credible explicit or implicit suggestion

in the cited references for a subsystem controller implemented as a single integrated circuit.

Had a single-IC implementation of a subsystem controller been obvious in view of the cited references, it would nothing short of astonishing that, by October 1999, in an extremely competitive, subsystem-controller commodity market, not one such implementation was available, produced, prototyped, described, or proposed. Applicants believe that without a clear showing of an explicit or implicit suggestion for a single-IC implementation of a subsystem controller, containing the claimed subsystem-controller elements, an obviousness-type rejection is not sustainable.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Michael B. Raynham et al.
Olympic Patent Works PLLC



Robert W. Bergstrom
Registration No. 39,906

Enclosures:
Postcard
Transmittal in duplicate
Extension of Time
Formal Drawings

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98194-0277
206.621.1933 telephone
206.621.5302 fax